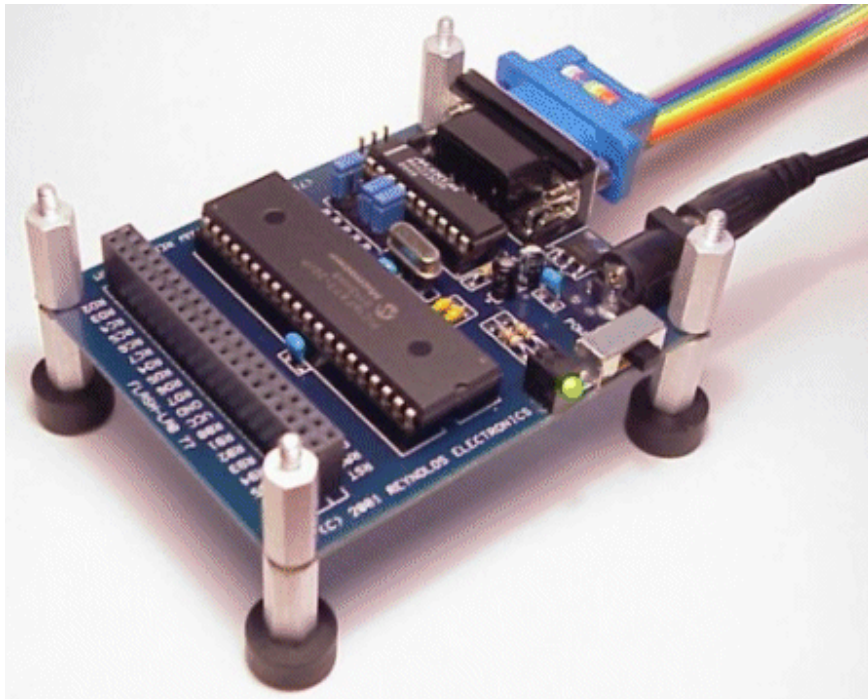


CISE LAB

Embedded Control Systems



CISE 414

Lab 1

Flashing LED

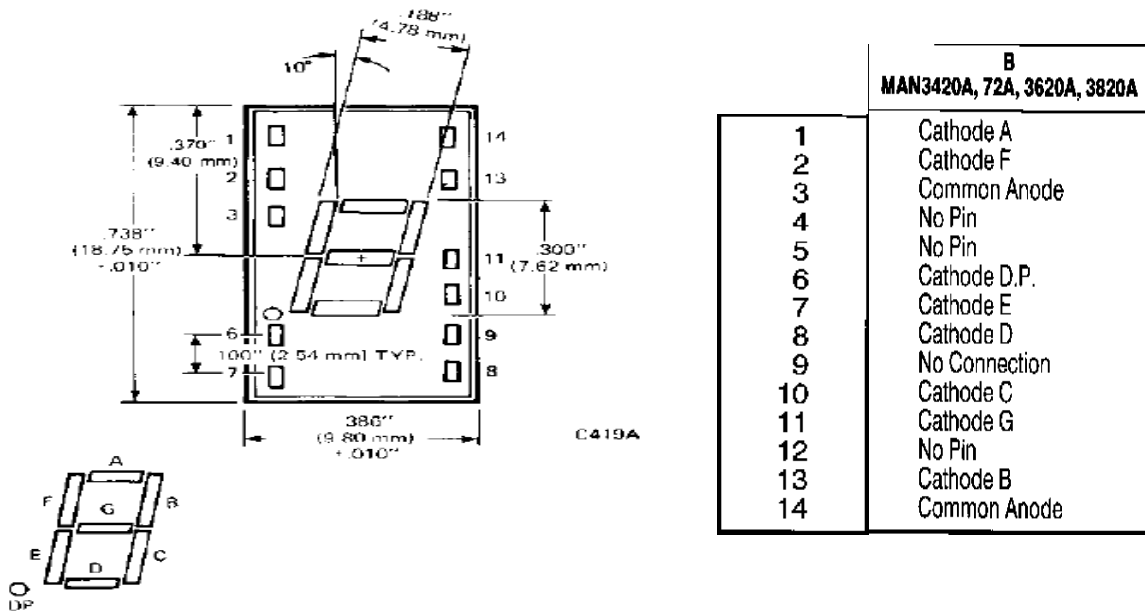
- 1- Find and open Microcode Studio on the computer desktop. MicroCode Studio is an easy to use code editor for microEngineering Labs PICBasic compilers. The main editor provides full syntax highlighting of your code with context sensitive keyword help and syntax hints. The code explorer allows you to automatically jump to include files, defines, constants, variables, aliases and modifiers, symbols and labels that are contained within your source code.
- 2- Write the following code:

```
*****
'* Name   : UNTITLED.BAS                               *
'* Author : [select VIEW...EDITOR OPTIONS]             *
'* Notice : Copyright (c) 2009 [select VIEW...EDITOR OPTIONS] *
'*       : All Rights Reserved                         *
'* Date   : 3/11/2009                                  *
'* Version : 1.0                                       *
'* Notes  :                                           *
'*       :                                           *
*****
DEfine OSC 40
LOOP: high portb.0
    pause 1000
    low portb.0
    pause 1000
    goto loop
end
```

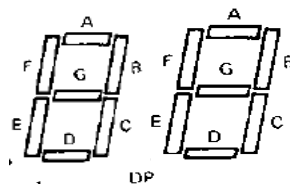
- 3- Make sure your Flashlab prototype board is connected to the PC using the RS232 9-bin connector and flat cable.**
- 4- Connect an LED with 220 Ohm resistor between Port B.0 and ground.**
- 5- Turn on your FlashLab board.**
- 6- From the Menu toolbar select “Compile and Program: option.**
- 7- Reset your FlashLab**
- 8- You will see now the LED is flashing. What is the on period and the off period?**
- 9- Modify your program so that the on period is 1 second, and the off period is 4 seconds.**
- 10- Show how to modify you're the circuit and program to have two LED, A and B, such that they alternate on and off.**

CISE LAB2

Seven Segment Display Encoder-Lab-2



1. Connect Pin 3 to VCC
2. Connect P1 microcontroller pins to cathodes of seven LED segment display pins through 150 ohms resistors. Configuration is given below.
P1.0=A, P1.1=B, P1.2=C, P1.3=D, P1.4=E, P1.5=F, P1.6=G
3. Logic level '0' turns on segment
4. Program for a circular counter 0-9. Put appropriate delay so that counting can be seen.
5. Now use two seven segment displays. Use P0 for other display. Now count 00-99



6. patterns to display digits are given as under

Number	Pattern (Hex)	Pattern(Binary) MSB..... LSB
1	79	0111 1001
2	24	0010 0100
3	30	0011 0000
4	19	0001 1000
5	12	0001 0010
6	03	0000 0011
7	78	0111 1000
8	00	0000 0000
9	18	0001 1000
0	40	0100 0000

1- Connect a 7-segment to port B of your microcontroller

2- Write, compile, and program the following code

```
*****
'* Name   : UNTITLED.BAS                               *
'* Author : [select VIEW...EDITOR OPTIONS]             *
'* Notice : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*       : All Rights Reserved                         *
'* Date   : 3/11/2008                                  *
'* Version : 1.0                                       *
'* Notes  :                                           *
'*       :                                           *
*****

    TRISB = %00000000
TRISC.0=1
LOOP: IF PORTC.0=1 THEN LOOP2
    PAUSE 5000
    PORTB=%00010001
    IF PORTC.0=1 THEN LOOP2
    PAUSE 5000
    PORTB=%01111101
LP1:   IF PORTC.0=1 THEN LP1
    PAUSE 5000
    PORTB=%00100011
LP2:   IF PORTC.0=1 THEN LP2
    PAUSE 5000
    PORTB=%00101001
    LP3:   IF PORTC.0=1 THEN LP3
    PAUSE 5000
    PORTB=%01001101
    LP4:   IF PORTC.0=1 THEN LP4
    PAUSE 5000
    PORTB=%10001001
    LP5:   IF PORTC.0=1 THEN LP5
    PAUSE 5000
    PORTB=%10000001
    LP6:   IF PORTC.0=1 THEN LP6
    PAUSE 5000
    PORTB=%00111101
    LP7:   IF PORTC.0=1 THEN LP7
    PAUSE 5000
    PORTB=%00000001
    LP8:   IF PORTC.0=1 THEN LP8
    PAUSE 5000
    PORTB=%00001101
    LP9:   IF PORTC.0=1 THEN LP9
    PAUSE 5000
```

```
GOTO LOOP
```

```
LOOP2:
```

```
    IF PORTC.0=0 THEN LOOP
```

```
    PAUSE 5000
```

```
GOTO LOOP
```

```
=====XXXXXXXXX=====
```

3- Show how to make the 7-Segment display Arabic numbers

CISE 414 LAB 3

Interfacing a Keyboard

```
*****
'* Name   : UNTITLED.BAS                               *
'* Author : [select VIEW...EDITOR OPTIONS]             *
'* Notice : Copyright (c) 2009 [select VIEW...EDITOR OPTIONS] *
'*       : All Rights Reserved                         *
'* Date   : 3/25/2009                                  *
'* Version : 1.0                                       *
'* Notes  :                                           *
'*       :                                           *
*****
DEFINE OSC 40
TRISC=%00000111
TRISA=0
TRISB=0
TRISD=0
```

```
main2:
high portc.4
low portc.5
low portc.6
if portc.0==1 then
PORTB="6"
PORTA.0=1
PORTB=%10111111
;gosub lat
endif

if portc.1==1 then
PORTB="5"
PORTA.1=1
PORTB=%10111011
;gosub lat
endif

if portc.2==1 then
```

```
PORTB="4"  
PORTB=%11011001  
PORTA.2=1  
;gosub lat  
endif
```

```
;%%%%%%%%%%
```

```
high portc.5  
low portc.4  
low portc.6  
if portc.0==1 then  
PORTB="9"  
PORTA.0=1  
PORTB=%11111001  
;gosub lat  
endif
```

```
if portc.1==1 then  
PORTB="8"  
PORTA.1=1  
PORTB=%11111111  
;gosub lat  
endif
```

```
if portc.2==1 then  
PORTB="7"  
PORTB=%11110001  
PORTA.2=1  
;gosub lat  
endif
```

```
;%%%%%%%%%%
```

```
high portc.6  
low portc.4  
low portc.5  
if portc.0==1 then  
PORTB="3"  
PORTA.0=1  
PORTB=%11101011  
;gosub lat  
endif
```

```
if portc.1==1 then
```



```
PORTB="2"  
PORTA.1=1  
PORTB=%11101110  
;gosub lat  
endif
```

```
if portc.2==1 then  
PORTB="1"  
PORTB=%11000001  
PORTA.2=1  
;gosub lat  
Endif
```

CISE 414 LAB4

Interfacing LCD display and Keyboard

```
*****  
* Name : UNTITLED.BAS *  
* Author : [select VIEW...EDITOR OPTIONS] *  
* Notice : Copyright (c) 2009 [select VIEW...EDITOR OPTIONS] *  
* : All Rights Reserved *  
* Date : 3/25/2009 *  
* Version : 1.0 *  
* Notes : *  
* : *  
*****
```

```
DEFINE OSC 40  
TRISC=%00001111  
TRISA=0  
TRISB=0  
TRISD=0
```

```
main:  
Pause 1500  
gosub init  
GOSUB shif  
GOSUB shif  
GOSUB shif  
GOSUB shif  
PORTB="W" ; send "W" on LCD  
GOSUB lat ; write on LCD  
PORTB="E"  
GOSUB lat  
PORTB="L"  
GOSUB lat  
PORTB="C"  
GOSUB lat  
PORTB="O"  
GOSUB lat  
PORTB="M"  
GOSUB lat  
PORTB="E"  
GOSUB lat  
GOSUB shif  
GOSUB shif  
GOSUB shif  
GOSUB shif
```

```
GOSUB shif
GOSUB shif
PORTB=" "
GOSUB lat
PORTB="T"
GOSUB lat
PORTB="O"
GOSUB lat
PORTB=" "
GOSUB lat
PORTB="C"
GOSUB lat
PORTB="I"
GOSUB lat
PORTB="S"
GOSUB lat
PORTB="E"
GOSUB lat
PORTB=" _ "
```

main1:

```
high portc.4
low portc.5
low portc.6
low portc.7
```

```
if portc.0==1 then
PORTB="7"
GOSUB lat
endif
```

```
if portc.1==1 then
PORTB="8"
gosub lat
endif
```

```
if portc.2==1 then
PORTB="9"
gosub lat
endif
```

;%%%%%%%%%%

**high portc.5
low portc.4
low portc.6
low portc.7**

**if portc.0==1 then
PORTB="4"
gosub lat
endif**

**if portc.1==1 then
PORTB="5"
gosub lat
endif**

**if portc.2==1 then
PORTB="6"
gosub lat
endif**

;%%%%%%%%%%

**high portc.6
low portc.4
low portc.5
low portc.7**

**if portc.0==1 then
PORTB="1"
gosub lat
endif**

**if portc.1==1 then
PORTB="2"
gosub lat
endif**

**if portc.2==1 then
PORTB="3"
gosub lat
endif**

goto main1

init:

pause 500

PORTB=%00111000 ;Function Set

LOW PORTD.7

LOW PORTD.1

HIGH PORTD.6

LOW PORTD.6

PAuse 500

PORTB=%00001110 ;Display ON/OFF

LOW PORTD.7

LOW PORTD.1

HIGH PORTD.6

LOW PORTD.6

PAuse 500

PORTB=%00001111 ;Display ON/OFF

LOW PORTD.7

LOW PORTD.1

HIGH PORTD.6

LOW PORTD.6

PAuse 500

PORTB=%00000001 ;clear display

LOW PORTD.7

LOW PORTD.1

HIGH PORTD.6

LOW PORTD.6

PAuse 500

PORTB=%00000110 ;clear display

LOW PORTD.7

LOW PORTD.1

HIGH PORTD.6

LOW PORTD.6

PAuse 500

RETURN

```
shif:
PORTB=%00010100 ;shift to right
LOW PORTD.7
LOW PORTD.1
HIGH PORTD.6
LOW PORTD.6
PAuse 500
RETURN
```

```
lat:
HIGH PORTD.7
LOW PORTD.1
HIGH PORTD.6
LOW PORTD.6
PAUSE 300
RETURN
```

```
;main2:
;high portc.4
;low portc.5
;low portc.6
;if portc.0==1 then
;PORTB="6"
;PORTA.0=1
;PORTB=%10111111
;gosub lat
;endif
```

```
;if portc.1==1 then
;PORTB="5"
;PORTA.1=1
;PORTB=%10111011
;gosub lat
;endif
;
;if portc.2==1 then
;PORTB="4"
;PORTB=%11011001
```

```
;PORTA.2=1
;gosub lat
;endif
```

```
;%%%%%%%%%
```

```
;high portc.5
;low portc.4
;low portc.6
;if portc.0==1 then
;PORTB='9'
;PORTA.0=1
;PORTB=%11111001
;gosub lat
;endif
```

```
;
;if portc.1==1 then
;PORTB='8'
;PORTA.1=1
;PORTB=%11111111
;gosub lat
;endif
```

```
;
;if portc.2==1 then
;PORTB='7'
;PORTB=%11110001
;PORTA.2=1
;gosub lat
;endif
```

```
;%%%%%%%%%
```

```
;high portc.6
;low portc.4
;low portc.5
;if portc.0==1 then
;PORTB='3'
;PORTA.0=1
;PORTB=%11101011
;gosub lat
;endif
```

```
;if portc.1==1 then
;PORTB='2'
;PORTA.1=1
```

```
;PORTB=%11101110  
;gosub lat  
;endif
```

```
;if portc.2==1 then  
;PORTB="1"  
;PORTB=%11000001  
;PORTA.2=1  
;gosub lat  
;endif //
```


CISE 414 LAB5

Interfacing Servomotors

```
*****
'* Name   : UNTITLED.BAS                               *
'* Author : [select VIEW...EDITOR OPTIONS]             *
'* Notice : Copyright (c) 2008 [select VIEW...EDITOR OPTIONS] *
'*       : All Rights Reserved                         *
'* Date   : 4/1/2008                                   *
'* Version : 1.0                                       *
'* Notes  :                                           *
'*       :                                           *
*****

    define OSC 20
period2 var word
limit var word
period var word
serv1_pos var byte
on Interrupt Goto myint
    serv1_pos = 30
    period = 800
    period2 = 40000
    INTCON = $C0
    pr2 = 250
    T2CON = $04
    PIR1.1 = 0
    PIE1.1 = 1
    TRISB.0 = 0
    PORTB.0 = 0
    TRISC.0 = 1
    TRISC.1 = 1

loop:
    GOTO loop
Disable
myint:
    period = period + 1
    limit = 500 + serv1_pos;
    if period > limit THEN
        PORTB.0 = 1
    endif
    if period > 800 THEN
        period = 0
        PORTB.0 = 0
```

```
endif
period2 = period2 + 1;
if period2 > 40000 THEN
    period2 = 0;
```

```
endif
PIR1.1 = 0
Resume
Enable
```

CISE 414 LAB 6

DC motor speed control using pwm

PART A)

Find attached the data sheet for D200 H-Bridge for DC motor drive.

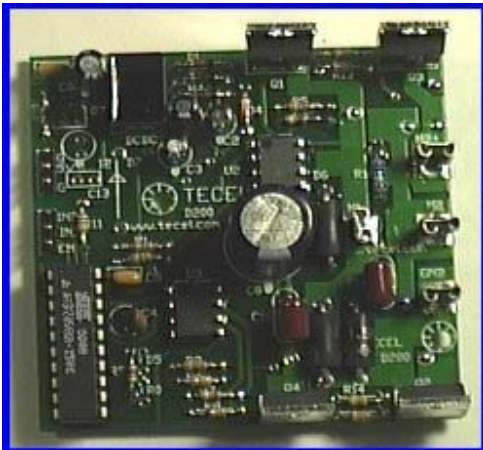
Use PortB as input with internal pull-up. And use PORTD.0 for pwm output
And PORTD (1,2) for output to drive IN1, and IN2 of the D200 board.

For pwm, use the pwm ProBasic command.

Connect PORTB to an 8-switch DIP package
DC Motors are controlled using a high power H-Bridge DC Motor Driver Board, D200

Model manufactured by Tecel Microcontrollers.

The below shown figure is D200 Multiwatt DC Motor Driver Board made by Tecel
Microcontrollers.



The Features of the H-bridge DC Motor Driver Board are as follows:

Power Semiconductors:	Four 110A MOSFETS
Rated Current:	10A Without Heat sinks
Max Current:	60A with Proper Heat sinks Installed

Board's Size: 2.5" x 2.5" (6.35cm x 6.35cm).

Components Type: 100% Solid State, No Relays

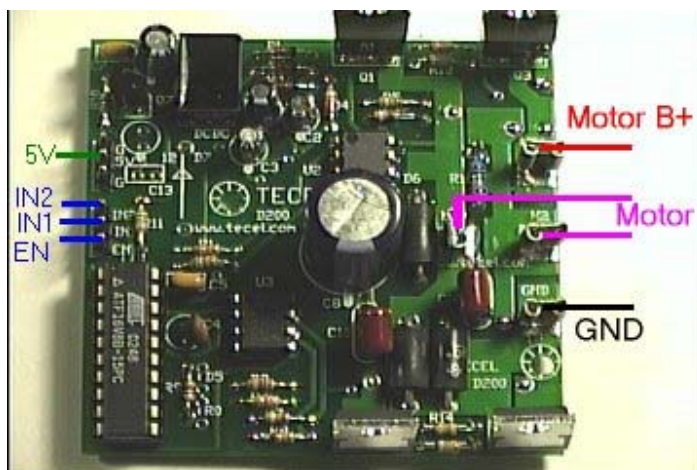
It has Filter Capacitors, VTS Diodes and snubber circuit ready on board. The voltage requirements for this driver board are 5 volts for the digital, and MOSFET Bias circuits and any voltage up to 55 volts for Motor B+. (USE 12 volt)

Besides the Motor B+, the D200 requires a regulated 5 Volt supply for 2 reasons:

- 1) **For the on board digital circuitry.** This should pose no problem, since all digital and microcontroller circuits, will be used in conjunction with the D200, are powered by a regulated 5 Volt supply.
- 2) **The on board DC-DC converter.** This circuitry generates 3 independent 10-volt supplies for the purpose of supplying the MOSFET's gates with 10 volts no matter what the motor B+ is at in the range of 0 to 55 volts.

Connections of DC Motor Driver board

The below shown is the connections from the Motor and Power supply to the driver board.



Motor B+ goes to the 24volts power supply and GND to ground.

The 2 pink lines goes to the Motor terminals.

The EN, IN1, IN2 are the control lines from the Processor ports.

The 5V supply required by the control board and the D200 is obtained by means of a 5V regulator.

=====

PART B)

Speed control using Potentiometer

Use PORTA.0 as an analog input AI

Connect AI to the center of a potentiometer (10 K or 5K) , on terminal of the potentiometer should be connected to a fixed resistance of the same value (10k or 5k) to VCC (5.0 V). Connect the other terminal of the potentiometer to ground.

Configure the AI control to read from CH0. Read the most significant 8 bits.

Use 0-127 as forward speed, 128 zero speed, 129-255 reverse motion.